

LT - type H

Pro-face

Digital
Human Machine Interface

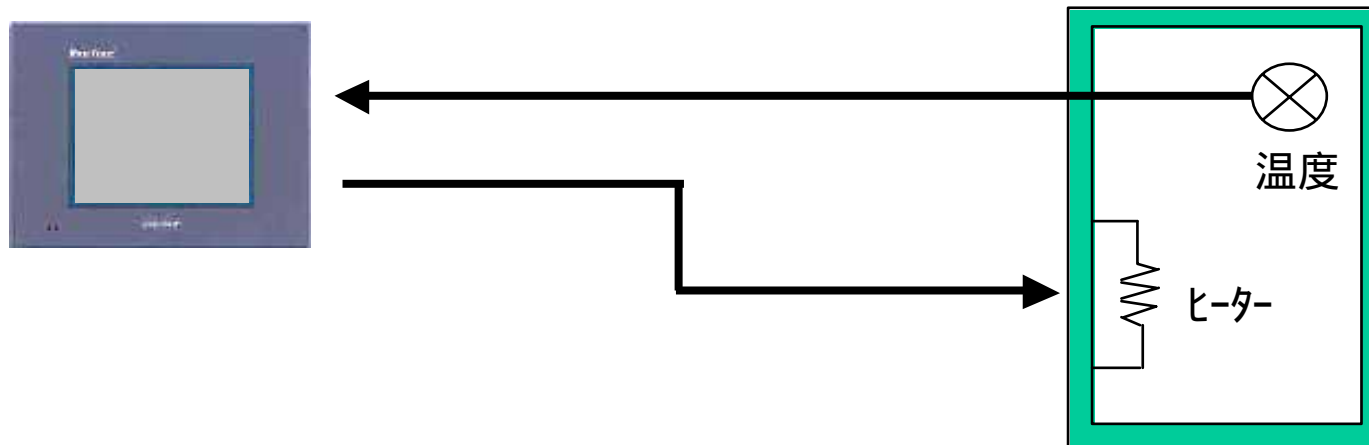
LT - type H

PID演算

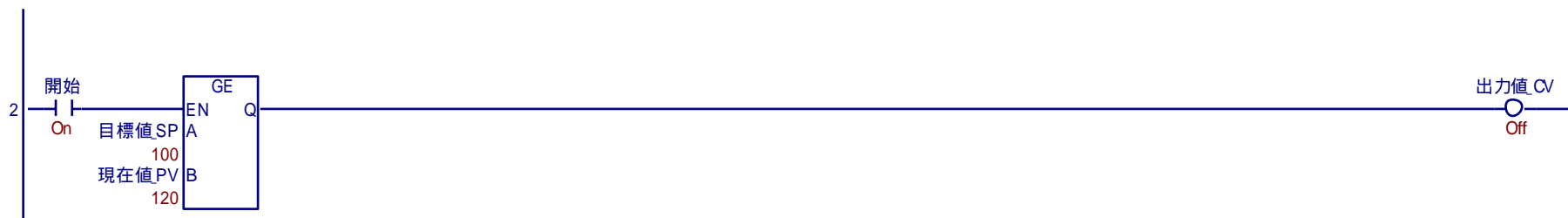
PID動作ってな～に～？

P動作（比例動作）

例えば、ヒーターをON/OFF制御し、目標とする温度に設定しようとしています。

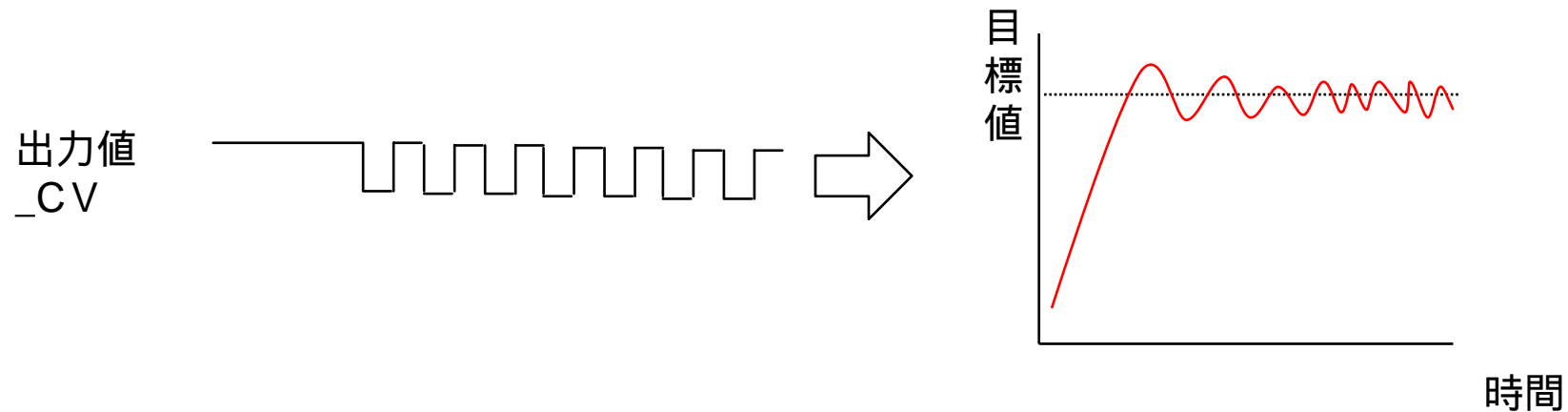


下図のようなラダーで、目標値に対して、高いか低いかでON/OFFを行ったとします。



PID動作ってな～に～？

そうすると、目標値でハンチングしてしまいました。

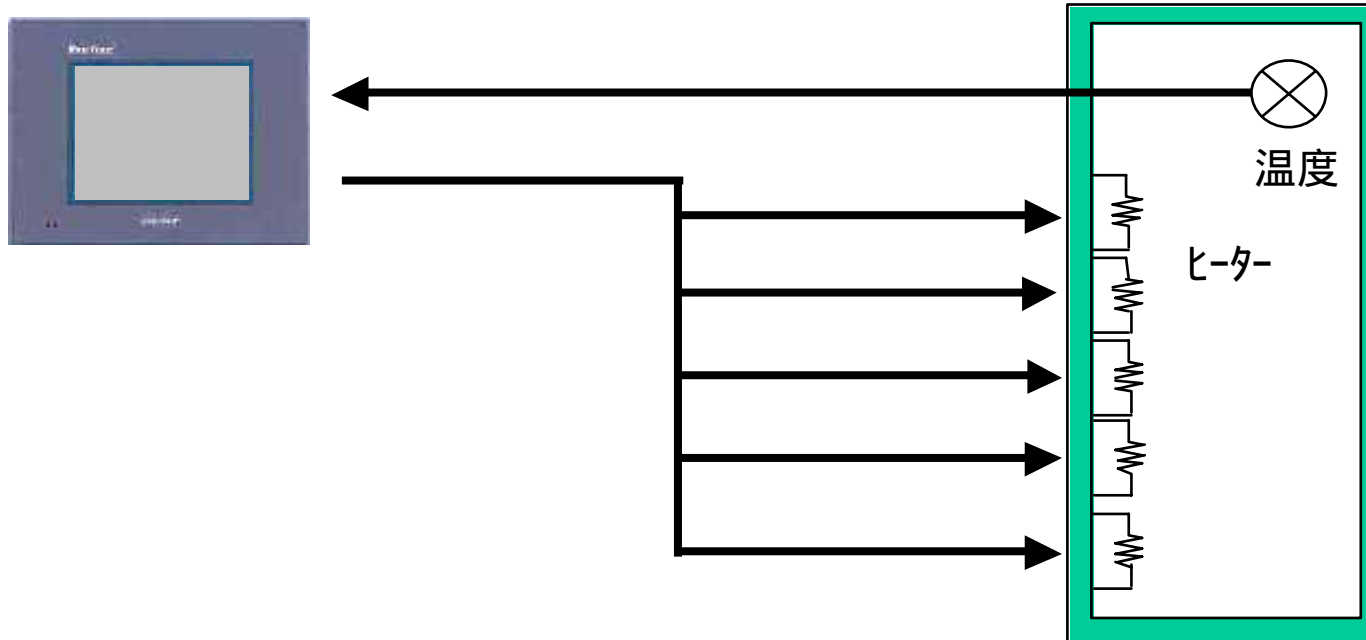


ハンチングの大きさは制御対象によって異なります。
ハンチングの振幅が小さく、それが許容範囲内であれば、ON/OFF制御を利用することは可能ですが、精密な制御で、ハンチングをなくそうとしたとき、限界があります。

PID動作ってな～に～？

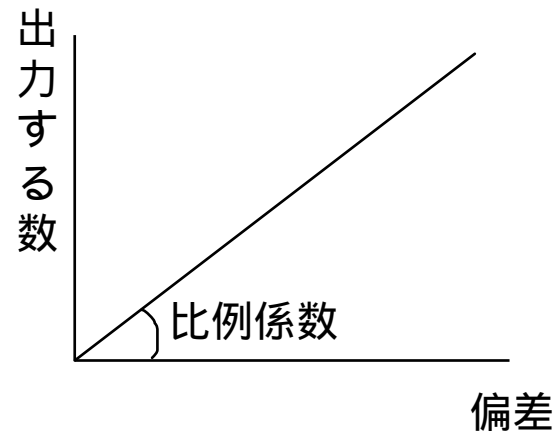
ON/OFF制御でハンチングが避けられないのは、操作変数が2値動作するからです。
(先程の場合、目標値 現在値でON、目標値 現在値でOFFの2つ)

もっと良い制御を行うには、操作変数の値をもっと多くとります。
つまり、目標値に近づいたら、小さく制御し、目標値から遠ざかったら、大きく制御する
というのが最適です。
先程を例にして、小さいヒーターを5つ取り付け、目標値と現在値の偏差からヒーターを
ONする数を変化させます！

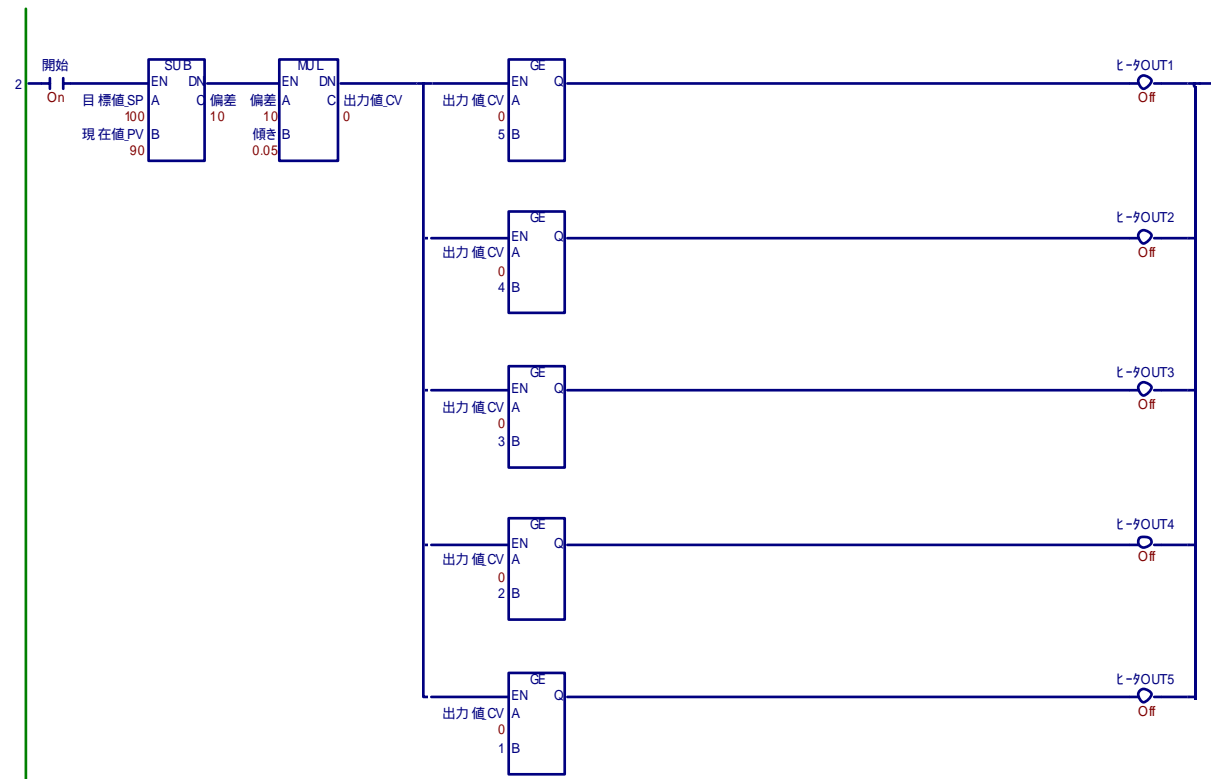


PID動作ってな～に～？

偏差に応じて、出力する数を制御するので、下図のような比例式になります。
ラダーにすると下図のようになります。

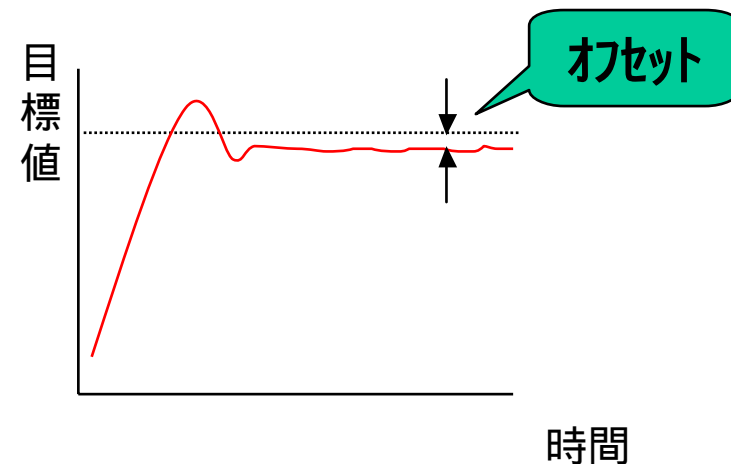


$$\begin{aligned} \text{出力する数} \\ &= \text{比例係数} \cdot \text{偏差} \\ &\quad + \text{初期出力数} \end{aligned}$$



PID動作ってな～に～？

前頁の制御を行うと、右図のように出力されました。



操作変数が偏差に比例することからこれを“比例動作”といい、“比例（プロポーショナル）”の頭文字を取って、**P動作**と言います。

P動作では、

- ・ 比例係数を大きくすると、小さな偏差で出力を大きく出すことができます。
- ・ 制御対象によっては、若干の振動はありますが、ハンチングを収めることができます。ただし、落ち着いた後に偏差が残ります。

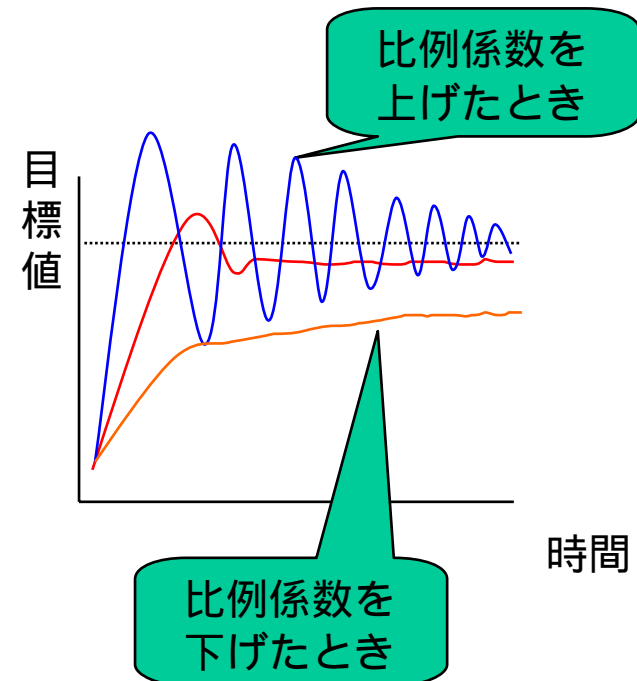
この定常的な偏差を**オフセット**と言います。

PID動作ってな～に～？

オフセットの値は比例係数を上げることで小さくできますが、大きくしすぎると、ハンチングを起こします。

つまり、P動作では、限界があり、オフセットは避けられない現象です。

(* 制御対象において、オフセット値が許容範囲内であれば、P動作は実用になります。)



PID動作ってな～に～？

I動作（積分動作）

I動作（積分動作）はオフセットをゼロにする能力をもっています。
 P動作と組み合わせること（PI動作）で、P動作の欠点を補うことができます。
 PI動作の式は、

$$\text{出力数} = \text{比例係数} \cdot (\text{偏差} + 1 / \text{積分時間} \cdot (\text{偏差}) dt)$$

となります。

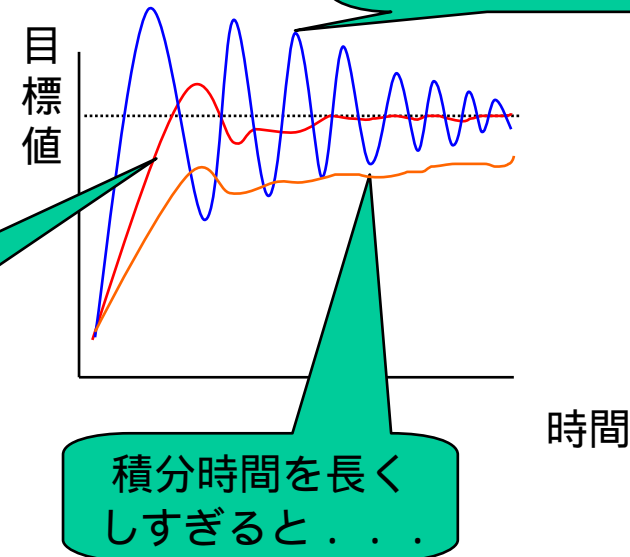
つまり、I動作とは、補正值です。

公式より積分時間を短くすれば、補正值が大きく（より効く）なり、
 長くすると、小さく（より効かない）なります。

制御対象によりますが、I動作についても、
 I動作を利かせすぎるとハンチングを起こします。
 逆に小さくしすぎると、P動作と同じになります。

* I動作を使うとP動作時より
 若干、応答（現在値）が
 遅れます。

適切に合わせると、
 オフセット分を補
 正できる。



P I D 動作ってな～に～？

D 動作（微分動作）

P 動作と組み合わせること（P D 動作）で、制御応答の改善を行うことができます。

P D 動作の式は、

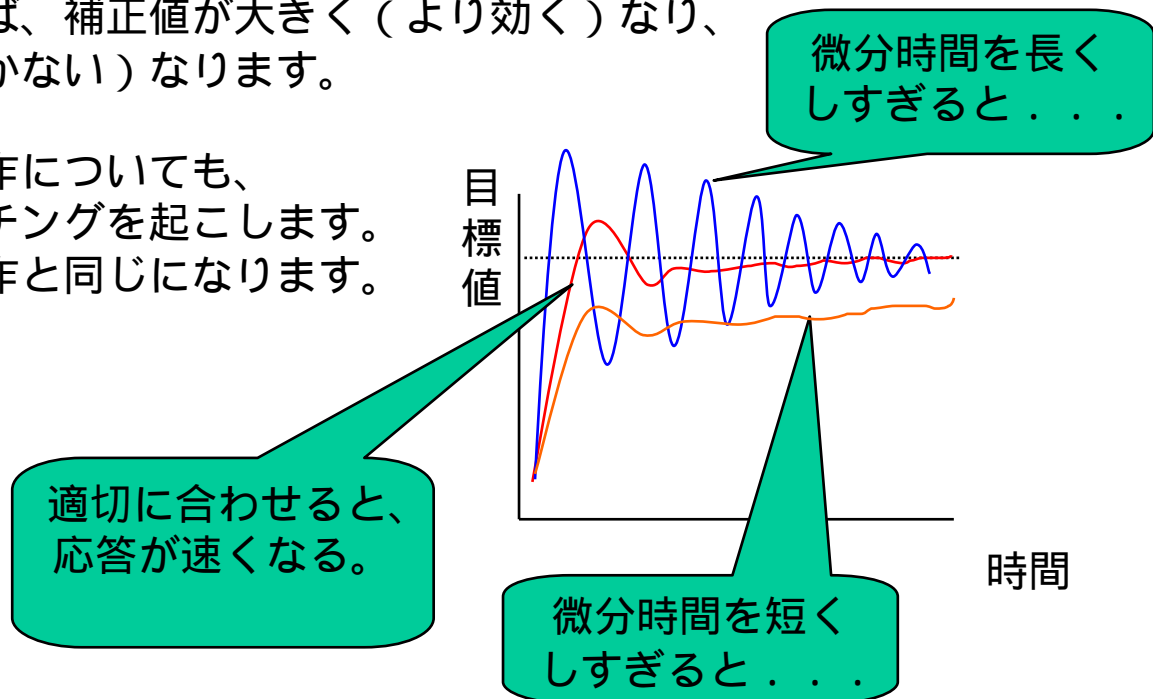
$$\text{出力数} = \text{比例係数} \cdot (\text{偏差} + \text{微分時間} \cdot d / d t (\text{偏差}))$$

となります。

つまり、D 動作は補正值です。

公式より微分時間を長くすれば、補正值が大きく（より効く）なり、短くすると、小さく（より効かない）なります。

制御対象によりますが、D 動作についても、D 動作を利かせすぎるとハンチングを起こします。逆に小さくしすぎると、P 動作と同じになります。



P I D 動作ってな～に～？

P I D 動作

P I D 制御の演算式は、以下のようになります。

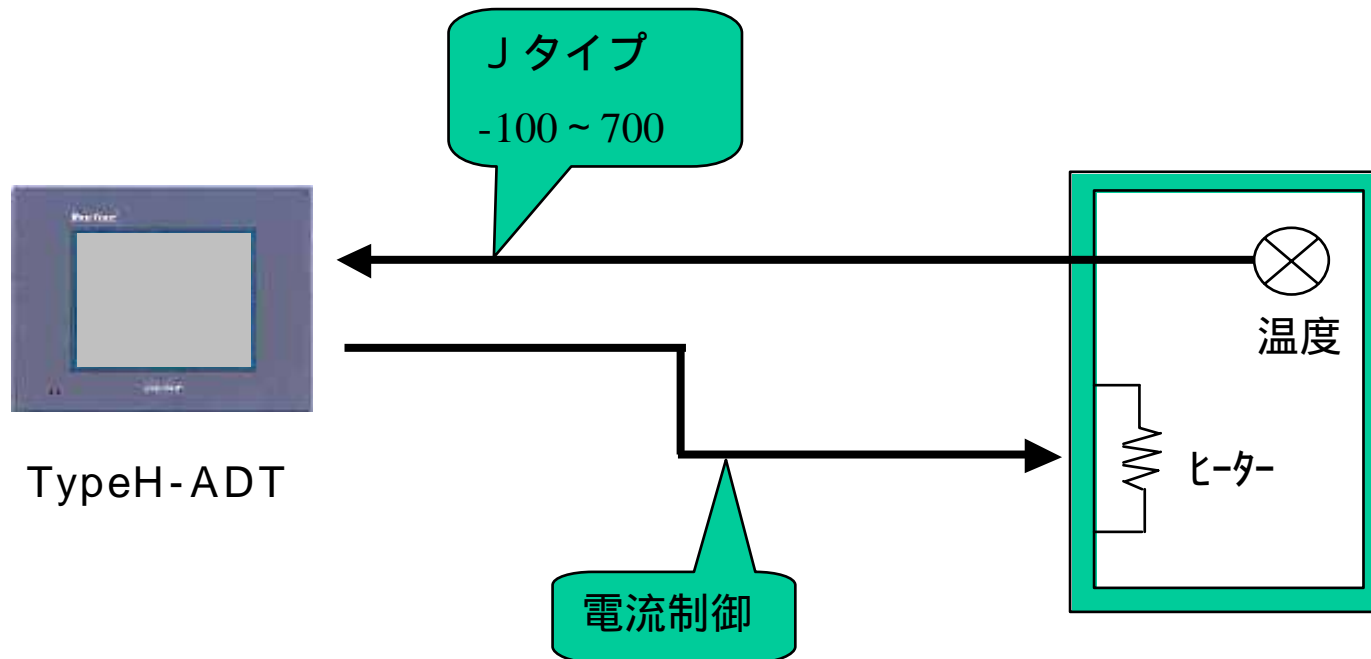
$$C V = \underbrace{(K \cdot e)}_{P \text{ 動作}} + \underbrace{\left(K \cdot \frac{1}{T_i} \cdot \int (e) dt \right)}_{I \text{ 動作}} + \underbrace{\left(K \cdot T_d \frac{d}{dt} (e) \right)}_{D \text{ 動作}}$$

C V : 出力値
K : 比例係数
e : 偏差
T i : 積分時間
T d : 微分時間

オフセットも無くしたい、制御応答も速くしたい。この両方を満足させるには P I D 動作を使用します。

PID命令はどうやって使うの？

下図のように、室内の温度をアナログ動作でヒーターを制御します。



P I D 命令はどうやって使うの？

まず、命令を配置する。

〔挿入〕 - 〔命令〕で命令の挿入ボックスが出ます。

そこで“PID”を選択し、挿入します。

まず、最初に P I D 命令の S P、P V、T B、C V に変数を設定します。



変数	内容	備 考
SP	目標値	設定する値を入力します。
PV	現在値	フィードバックの値。
TB	タバック	命令が導通していない場合、ここで設定された値が出力されます。
CV	出力値（操作量）	P I D 演算式より求められた数値。

PID命令はどうやって使うの？

〔PID / チューニング〕ダイアログボックス

PID命令をダブルクリックすると、下図のような〔PID / チューニング〕ダイアログボックスが開きます。

セットアップウィンドウで以下の設定を行います。



PID命令はどうやって使うの？

〔PID / チューニング〕ダイアログボックス

まず、範囲設定で“直接、100分率、単位設定”から選択します。

“直接上限 / 下限”、“設定上限 / 下限”、“出力制限範囲__上限値 / 下限値”、“積分回数範囲__上限値 / 下限値”を設定します。

範囲設定

直接	接続機器への入出力の値をそのまま使用する。
100分率	接続機器への入出力の値を比率（0～100%）で設定する。
単位設定	ユーザー設定による。
単位名	本チューニング画面に単位が付きます。
直接上限 / 下限	チューニング画面への入力に反映されます。
設定上限 / 下限	チューニング画面への入力に反映されます。

* “直接” / “100分率” / “単位”の中で“直接”で行うことをお勧めいたします。

PID命令はどうやって使うの？

〔PID / チューニング〕ダイアログボックス

直接の場合

直接上限 / 下限の範囲を使用する。

設定上限や出力制限範囲、チューニング画面での設定値などに影響があります。

出来る限り大きい値を入れておきましょう。

直接の場合、この値は左の直接上限 / 下限と同じになります。

出力側の上下限值を入れましょう！

出力側 : 0 ~ 4000 (0 ~ 20mA)

積分回数の上下限を設定します。

The screenshot shows the 'PID Tuning' dialog box with the following settings and callouts:

- Mode:** 直接 (Direct) / 100分率 (100% Rate)
- Unit Setting:** 単位設定 (Unit Setting) / 単位名 (Unit Name)
- Direct Limits:**
 - 直接上限 (Direct Upper Limit): 7000
 - 直接下限 (Direct Lower Limit): -1000
 - 設定上限 (Setting Upper Limit): 7000
 - 設定下限 (Setting Lower Limit): -1000
- Output Limit Range:**
 - 上限値 (Upper Limit): 4000
 - 下限値 (Lower Limit): 0
- Integral Count Range:**
 - 上限値 (Upper Limit): 4000
 - 下限値 (Lower Limit): 0
- Control:**
 - データ取得周期 (Data Acquisition Cycle): 1000 ms
 - 正動作 (SP-PV) / 逆動作 (PV-SP)
- Buttons:** OK, キャンセル, ヘルプ (Help), 更新 (Update), 元に戻す (Reset)
- Range:** 範囲: -1000 ~ 7000

PID命令はどうやって使うの？

〔PID / チューニング〕ダイアログボックス

100分率の場合
直接上限 / 下限を比率 (0 ~ 100%) で設定する。

100分率を選んだ時点で、0~100になります。

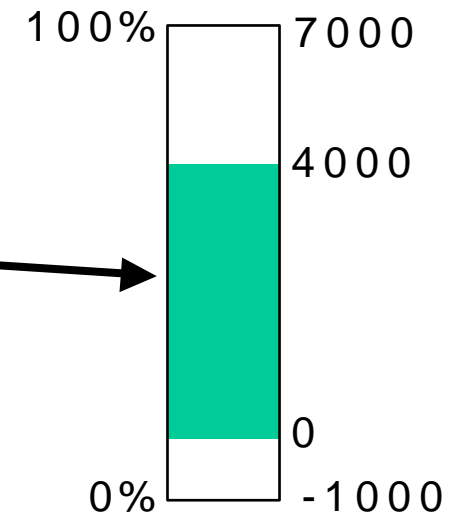
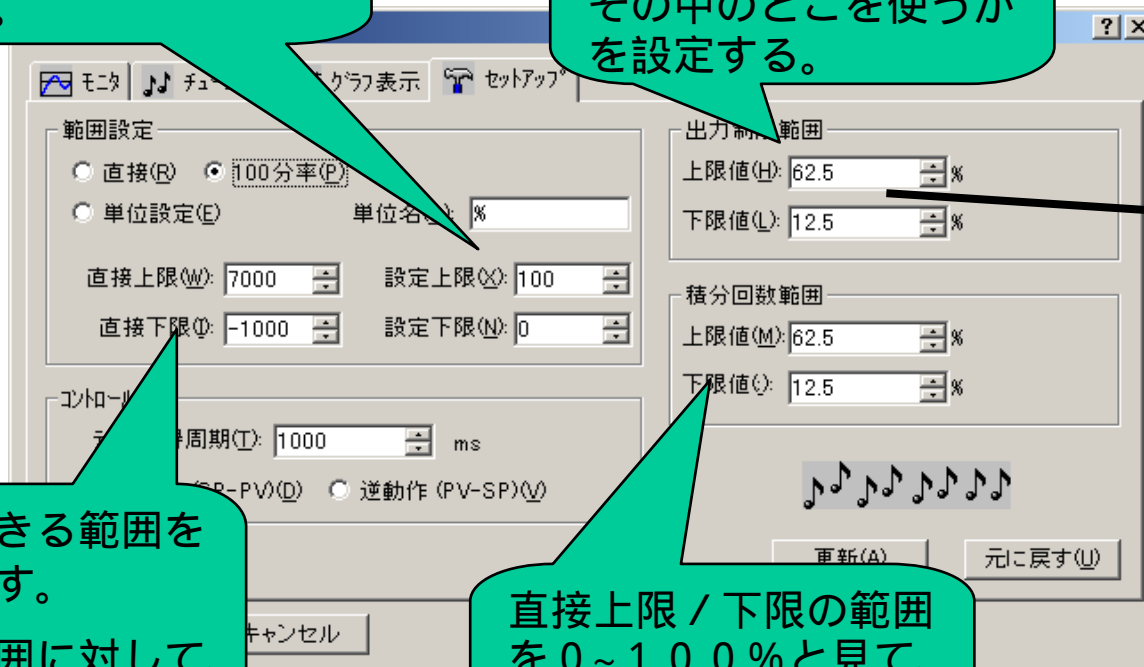
直接上限 / 下限の範囲を0~100%と見て、その中のどこを使うかを設定する。

設定できる範囲を決めます。

この範囲に対して、100分率されます。

直接上限 / 下限の範囲を0~100%と見て、その中のどこを使うかを設定する。

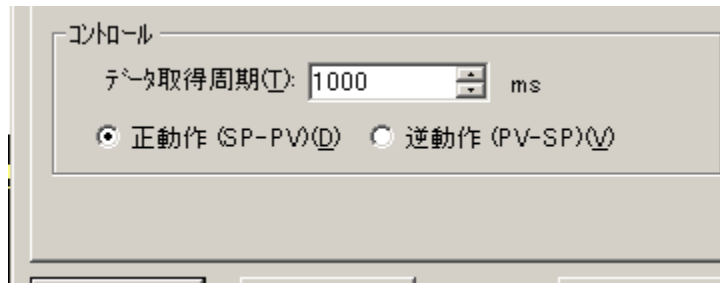
* こういう意味です。



PID命令はどうやって使うの？

〔PID / チューニング〕ダイアログボックス

データ取得周期とサンプリングの関係



$$EF_n = EF_{n-1} + \frac{T_{Loop}}{T_{Filter}} (E_n - EF_{n-1})$$

- EF : 偏差のフィルタ結果
- TLoop : ループ更新時間 (データ取得周期)
- TFilter : フィルタ時定数 (サンプリング時間)
- E : 偏差 (SP-PVまたはPV-SP)
- n : サンプリング回数

* データ取得周期とは、PID演算の実行周期です。

上記式は、偏差に対するフィルタリング演算です。

サンプリング時間 (分母) を大きくすることで、偏差に乗るノイズの影響を小さくすることができます。

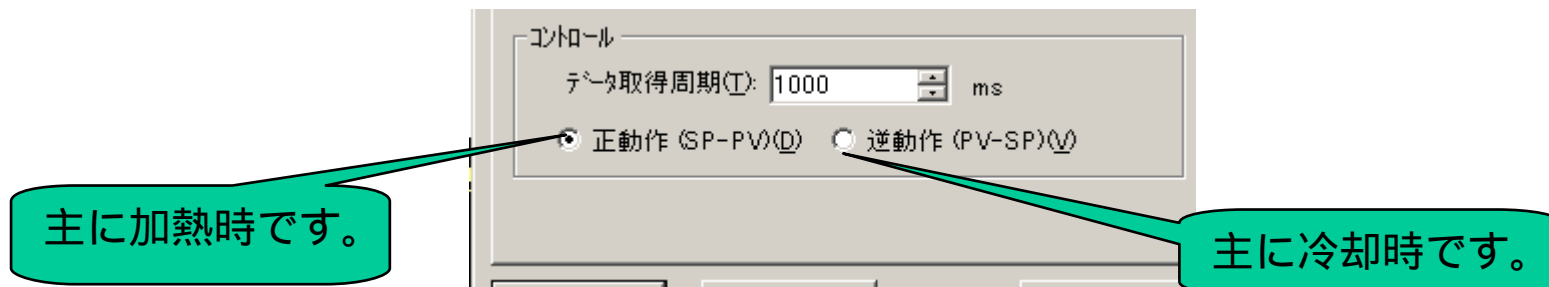
サンプリング時間を0にすると、このフィルタリング処理は行われません。

PID命令はどうやって使うの？

〔PID / チューニング〕ダイアログボックス

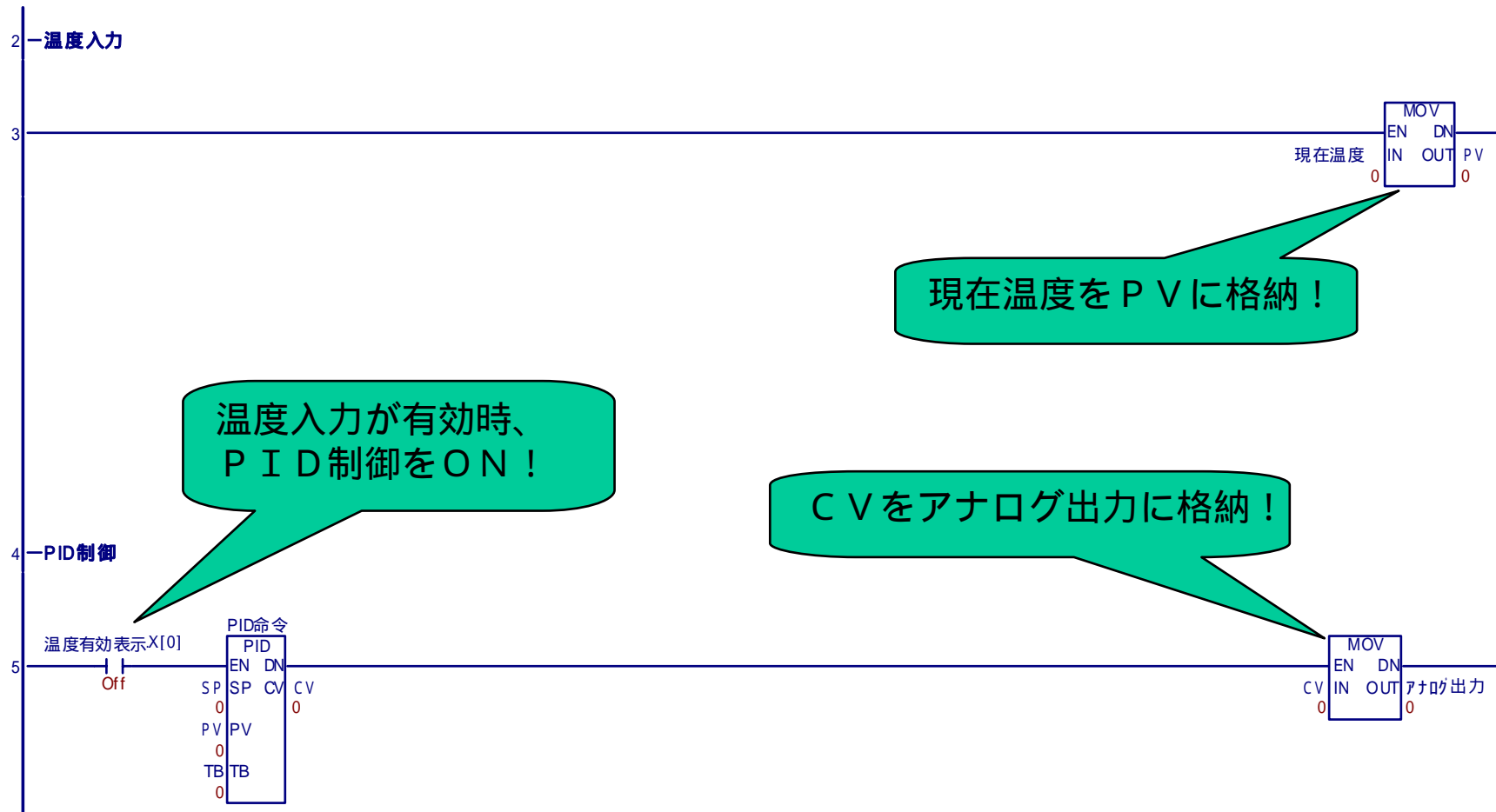
正動作 (SP - PV) と逆動作 (PV - SP) の選択

- “正動作 (SP - PV)” …… 現在値 (PV) を目標値 (SP) に上げるとき、こちらを選択。(加熱時)
- “逆動作 (PV - SP)” …… 現在値 (PV) を目標値 (SP) に下げるとき、こちらを選択。(冷却時)



*長年、温調器をお使いいただいている方へ
他の温調器では、加熱時 逆動作、冷却時 正動作ですが、弊社は意味が反対です。

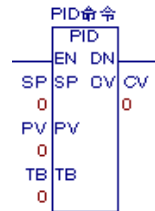
P I D 命令はどうやって使うの？ ラダーを作成する。



P I D 命令はどうやって使うの？

コントロールブロック変数

P I D 命令に変数を割り付けると、その変数には、下表の要素数 7 の配列が自動的に割り付けられます。要素 [0] はステータス、要素 [1] ~ [6] は P I D 制御の微調整を行うことができます。



P I D 命令 [0]

P I D 命令 [6]

要素番号	詳細	備考	
0	ビット0	モード切替フラグ	PID命令が導通すると、ON。 導通していないと、OFF。
	ビット1	P I D 命令処理の完了フラグ	CV値を出力するタイミングでONする。
	ビット2	P I D 処理無効範囲フラグ	無効範囲内でONする。
	ビット3	出力値の上限オーバー	チューニング画面のセットアップの 出力制限範囲の上限 / 下限を超えたもし くは、下回ったとき、それぞれがONする。
	ビット4	出力値の下限オーバー	
	ビット5	積分回数処理オーバー	チューニング画面のセットアップの 積分回数範囲外の時、ONする。

PID命令はどうやって使うの？ コントロールブロック変数

要素番号	詳細	備考
1	比例係数	
2	毎分あたりの積分回数	
3	1回あたりの微分時間	
4	PID処理無効範囲	
5	オフセット	
6	サンプリング時間	

チューニング画面と、コントロールブロック変数の関係
 コントロールブロック変数 [1] ~ [3] は、チューニング画面では、
 1 / 1 0 0 0 されて反映されます。

コントロールブロック変数[1]

1 2 3 4



コントロールブロック変数[1]に格納された
 値は、1 / 1 0 0 0 されて、チューニング
 画面に反映されます。

タイバック(T):	0	▼
比例係数(K):	1.234	▼
積分回数(R):	0	▼ 回
微分時間(E):	0	▼ 分
処理無効範囲(D):	0	▼
オフセット(O):	0	▼

PID命令はどうやって使うの？

コントロールブロック変数

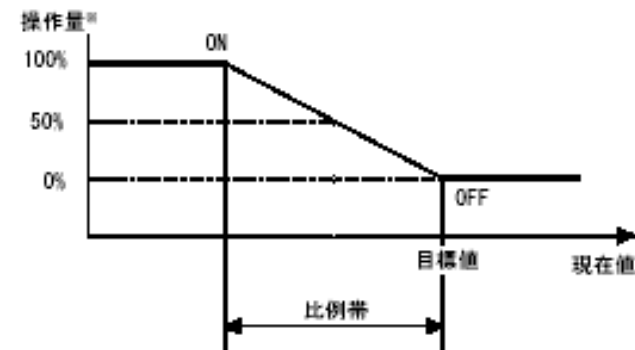
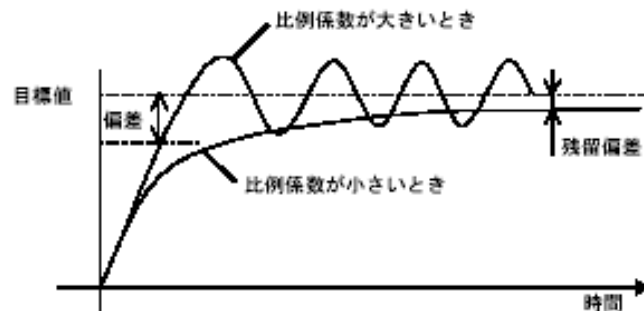
比例係数

比例係数は、大きくしすぎると、目標到達時間が短くなりますが、オーバーシュートの可能性があります。

逆に小さくしすぎると、残留偏差が大きくなる原因になります。

比例帯について

PID命令の比例制御において一般的温調メーカーでは比例帯の範囲内にて目標値が設定されておりますが、これはオフセットがデフォルト50%の設定となっております。一般的な温調器は当然温度制御が目的なので温度制御に適したオフセットが最初から設定されておりますが汎用的なPID制御を仕様としているLTのPID命令ではオフセットのデフォルトが0となっております。



※ 操作量：単位時間あたりの出力量

$$\text{比例帯} = 100 / \text{比例係数} [\%]$$

PID命令はどうやって使うの？

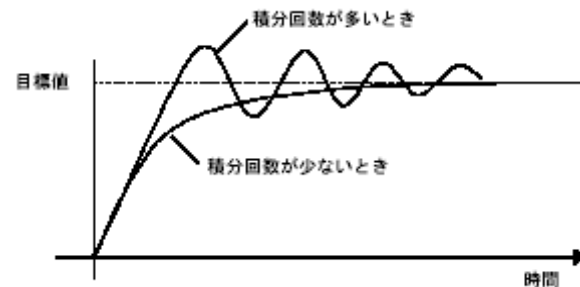
コントロールブロック変数

積分回数

積分回数は、1分あたり何回積分を行うかを設定します。

積分回数を多くしすぎると、目標到達時間は短くなりますが、オーバーシュート、ハンティングの原因となります。

積分回数を少なくしすぎると、目標到達時間が長くなります。



秒で設定した場合

120秒と設定したい

チューニング画面より

積分回数(R): 0.5 回 / 分
微分時間(F): 0 分

コントロールブロック変数[2]より

500

30秒と設定したい

積分回数(R): 2 回 / 分

2000

PID命令はどうやって使うの？

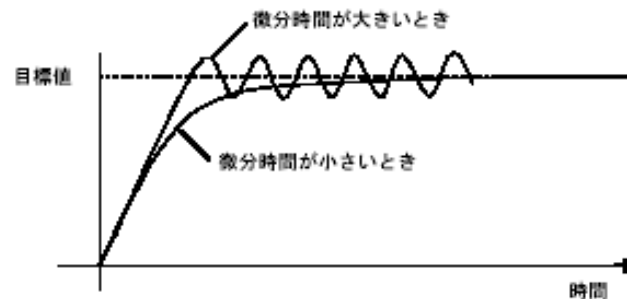
コントロールブロック変数

微分時間

微分時間を設定します。

微分時間を大きくしすぎると、外乱に対して復旧時間は短くなりますが、オーバーシュート、ハンティングの原因となります。

微分時間を小さくしすぎると、外乱に対して復旧時間が長くなります。



秒で設定した場合

120秒と設定したい

チューニング画面より

微分時間(D): 分

コントロールブロック変数[3]より

30秒と設定したい

微分時間(D): 分

PID命令はどうやって使うの？

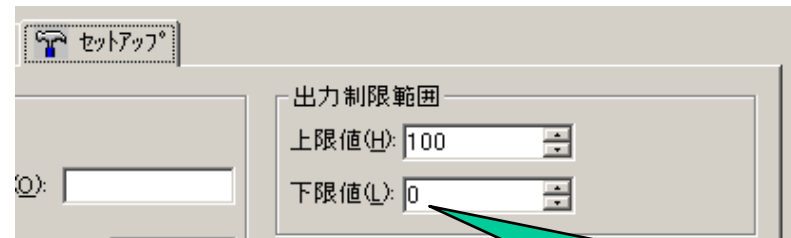
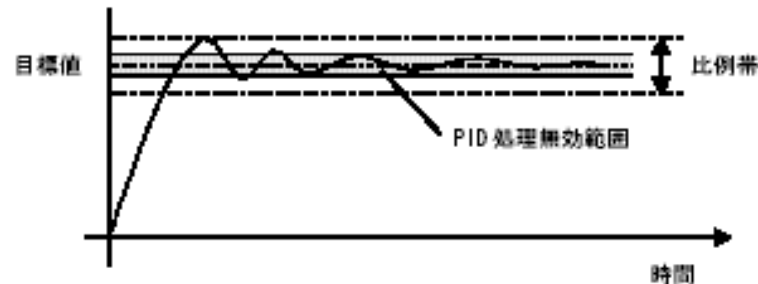
コントロールブロック変数

処理無効範囲

PID制御を行わない範囲を設定します。

目標値に現在値が達したとき、この動作を行います。(コントロールブロック変数 [0] のビット 2 のPID処理無効範囲フラグがON)

この範囲に入ったとき、チューニング画面の出力制限範囲の下限值が出力されます。



設定方法を以下の例により、説明します。

例) 温度 50.0 を ±10.0 無効範囲に設定する場合

チューニング画面から

目標値(S):	500
タイバック(T):	0
比例係数(K):	65
積分回数(R):	0 回 / 分
微分時間(E):	0 分
処理無効範囲(D):	100

コントロール変数 [4] へ書き込む時

100

PID命令はどうやって使うの？

コントロールブロック変数

設定方法を以下の例により、説明します。

例) 温度50 を±10 無効範囲に設定する場合

チューニング画面から

目標値(S):	<input type="text" value="50"/>
タイバック(T):	<input type="text" value="0"/>
比例係数(K):	<input type="text" value="65"/>
積分回数(R):	<input type="text" value="0"/> 回 / 分
微分時間(E):	<input type="text" value="0"/> 分
処理無効範囲(D):	<input type="text" value="10"/>

コントロール変数 [4] へ書き込む時

10

例より、処理無効範囲の重みは、実際に使用する目標値 (S P) の重みに対して、変化します。

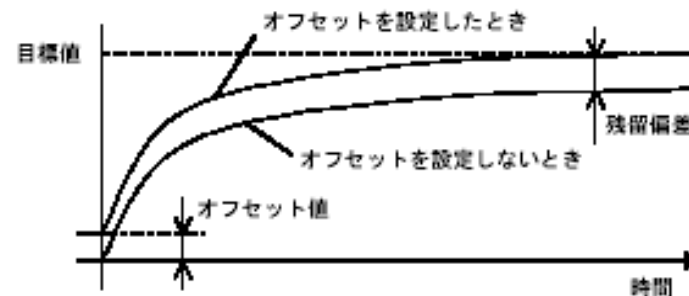
PID命令はどうやって使うの？

コントロールブロック変数

オフセット

例えば、目標値 (SP) + 40.0 に対して、現在値 (PV) が + 39.0 であったとします。ここで、オフセットに “ 10 ” と入力すると、現在値を下駄上げします。(目標値を + 41.0 で制御している状態になる)

オフセットの重みも、実際に使用する目標値 (SP) の重みに対して、変化します。



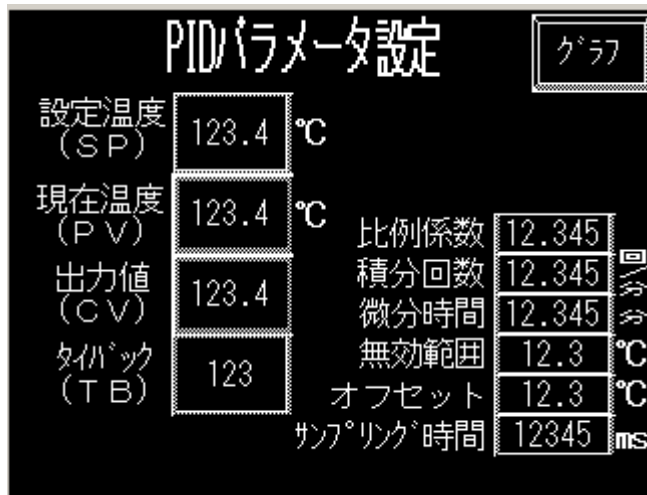
サンプリング時間

16ページを参照してください。

PID命令はどうやって使うの？

タッチパネルから設定をする！

下図のような画面を作ります。



設定画面



グラフ表示画面

各部品の詳細設定を次頁以降に記します。

PID命令はどうやって使うの？

設定画面を作る！

設定温度 (SP)・・・設定値表示器設定

設定値表示器設定 [KD_001]

情報 表示形式 形/カラー 警報設定 拡張

ワードアドレス SP

起動ビットアドレス 000000

ホップアップキーボード使用

アラウザ...

設定値表示器設定 [KD_001]

情報 表示形式 形/カラー 警報設定 拡張

絶対 相対 文字列

符号 +/- 四捨五入

表示・書込みデータ形式

<input checked="" type="radio"/> 16ビット Dec	<input type="radio"/> 32ビット Dec
<input type="radio"/> Hex	<input type="radio"/> Hex
<input type="radio"/> BCD	<input type="radio"/> BCD
<input type="radio"/> Oct	<input type="radio"/> Bin
<input type="radio"/> Bin	<input type="radio"/> Float

表示桁数 4 文字サイズ 1 x 1 (f)

小数点桁数 1

設定値表示器設定 [KD_001]

情報 表示形式 形/カラー 警報設定

警報表示

警報範囲

最小値 0

最大値 1000

警報カラー

Fg	<input type="checkbox"/> Black	<input type="checkbox"/> Blue	<input type="checkbox"/> Green	<input type="checkbox"/> Cyan	<input type="checkbox"/> Red	<input type="checkbox"/> Magenta	<input type="checkbox"/> Yellow	<input type="checkbox"/> White	<input type="checkbox"/> Blk
Bg	<input type="checkbox"/> Black	<input type="checkbox"/> Blue	<input type="checkbox"/> Green	<input type="checkbox"/> Cyan	<input type="checkbox"/> Red	<input type="checkbox"/> Magenta	<input type="checkbox"/> Yellow	<input type="checkbox"/> White	<input type="checkbox"/> Blk

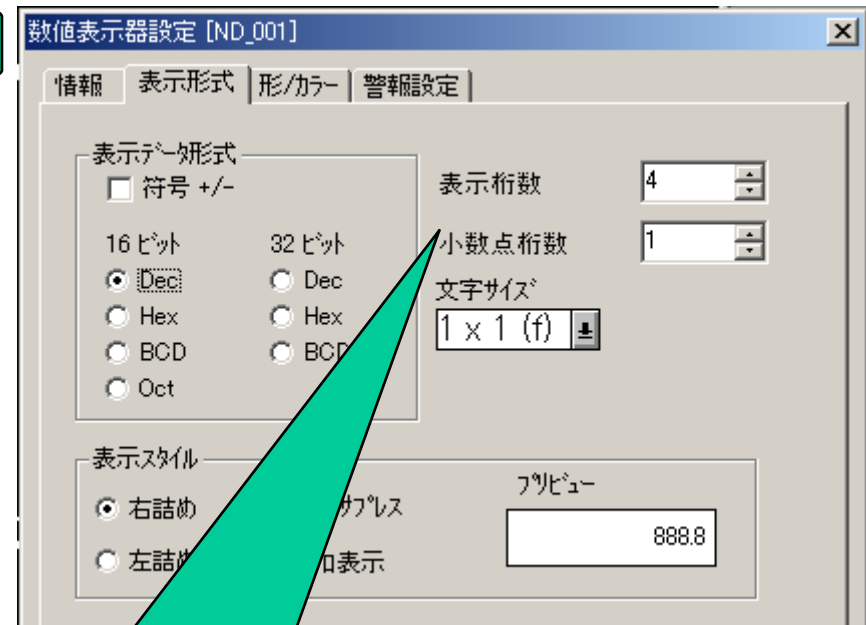
0.0 ~ 100.0 まで表示するとした場合。

Pt100入力は、-50.0 ~ +400.0 で、少数第1位まで表示できるので！

PID命令はどうやって使うの？

設定画面を作る！

現在温度 (PV)・・・数値表示器設定

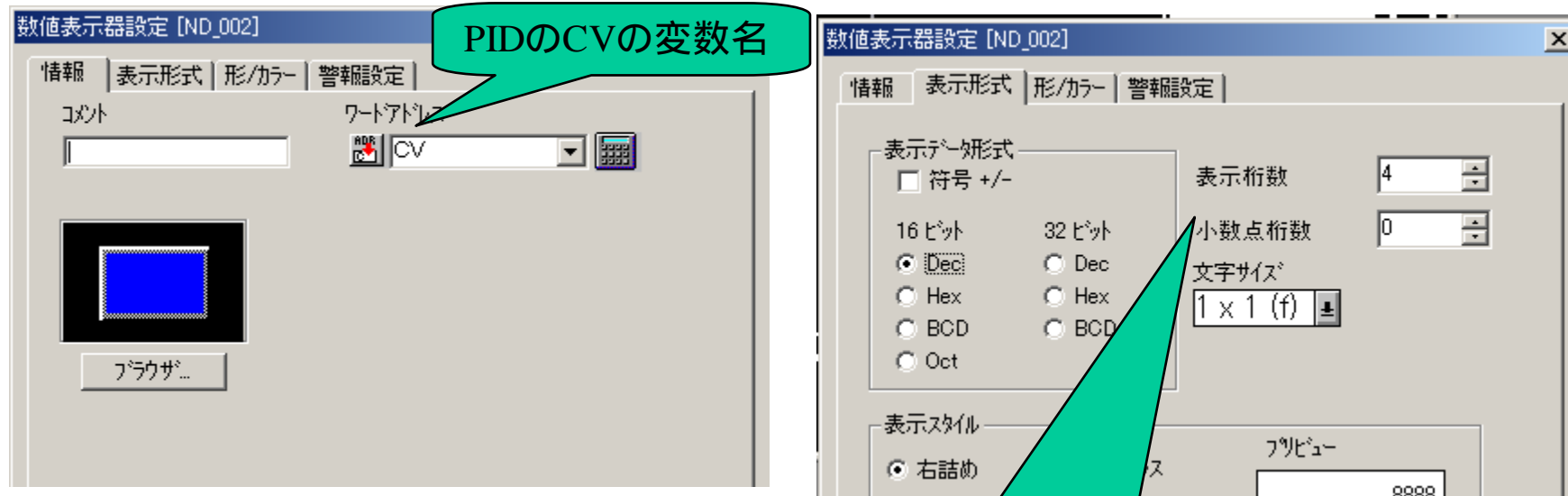


Pt100入力は、-50.0 ~ +400.0 で、少数第1位まで表示できるので！

PID命令はどうやって使うの？

設定画面を作る！

出力値（CV）・・・数値表示器設定



PIDのCVの変数名

CVの表示桁数。今回、アナログ出力を0～4095なので、4桁

PID命令はどうやって使うの？

設定画面を作る！

タイバック (TB)・・・設定値表示器設定

設定値表示器設定 [KD_002]

情報 | 表示形式 | 形/カラー | 警報設定 | 拡張

コメント

ワードアドレス: TB

起動ビットアドレス: 000000

ホップアップキーボード使用

アラウザ...

設定値表示器設定 [KD_002]

情報 | 表示形式 | 形/カラー | 警報設定 | 拡張

絶対 相対 文字列

符号 +/- 四捨五入

表示・書込みデータ形式

16ビット: Dec Hex BCD Oct Bin

32ビット: Dec Hex BCD Bin Float

表示桁数: 4

小数点桁数: 0

文字サイズ: 1 x 1 (f)

設定値表示器設定 [KD_002]

情報 | 表示形式 | 形/カラー | 警報設定 | 拡張

警報表示

警報範囲

最小値: 0

最大値: 4000

警報カラー

Fg: Black Blue Green Cyan Red Magenta Yellow Blk

Bg: Black Blue Green Cyan Red Magenta Yellow Blk

設定値表示器設定 [KD_002]

情報 | 表示形式 | 形/カラー | 警報設定 | 拡張

表示桁数: 4

小数点桁数: 0

文字サイズ: 1 x 1 (f)

PID命令はどうやって使うの？

設定画面を作る！

積分回数・・・設定値表示器設定

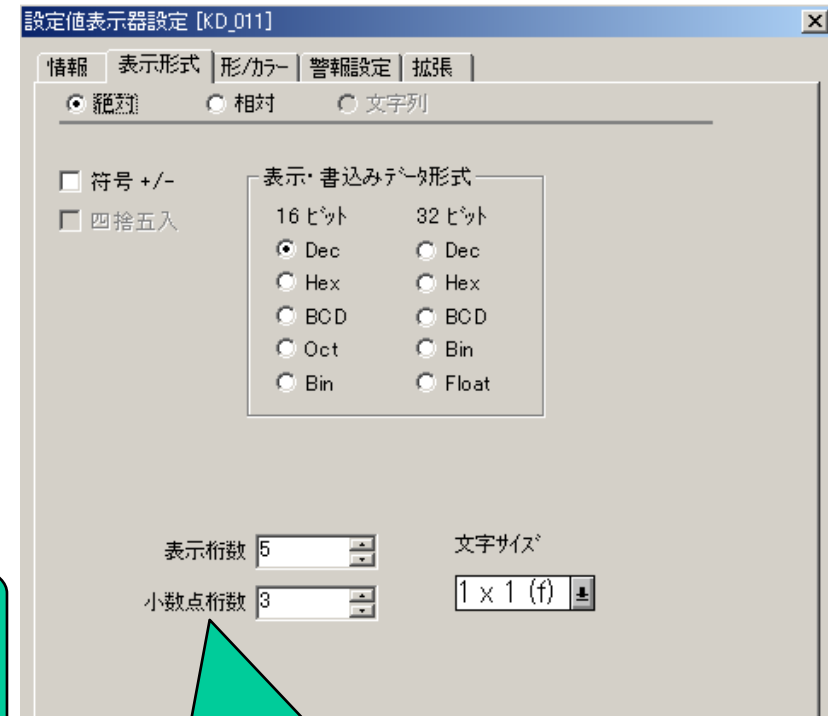
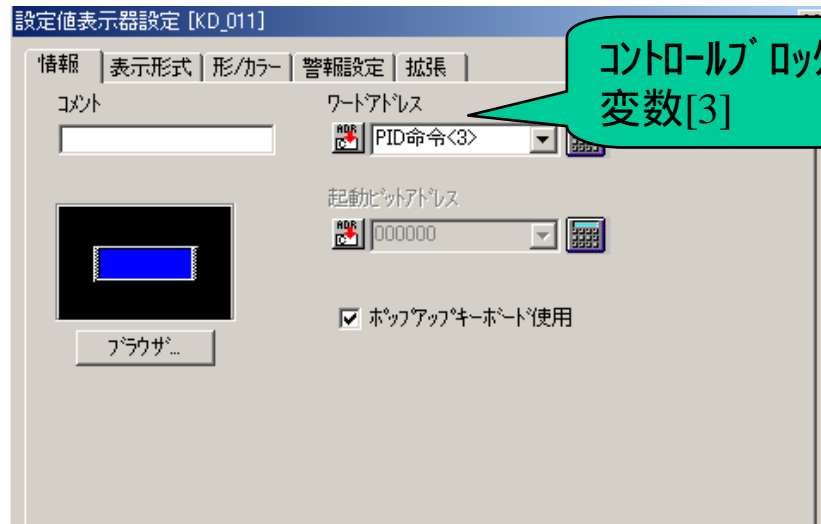
The screenshot shows the '設定値表示器設定 [KD_010]' window with several callouts:

- Control Lock Variable [2]:** A callout points to the 'ワードアドレス' (Word Address) field, which is set to 'PID命令<2>'.
- Integral Gain Range Setting:** A callout points to the '警報範囲' (Warning Range) section, where the '最小値' (Minimum) is 0 and the '最大値' (Maximum) is 65535. A note states: '* 65535(65.535)は任意' (65535/65.535 is arbitrary).
- Scaling:** A callout points to the '表示桁数' (Number of Digits) set to 5 and '小数点桁数' (Number of Decimal Digits) set to 3, with a note: 'チューニング画面に対して、1/1000スケールなので！' (For the tuning screen, it's a 1/1000 scale!).

PID命令はどうやって使うの？

設定画面を作る！

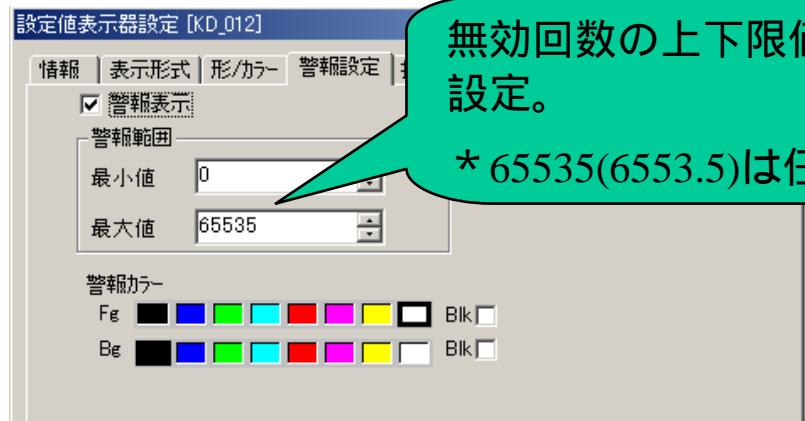
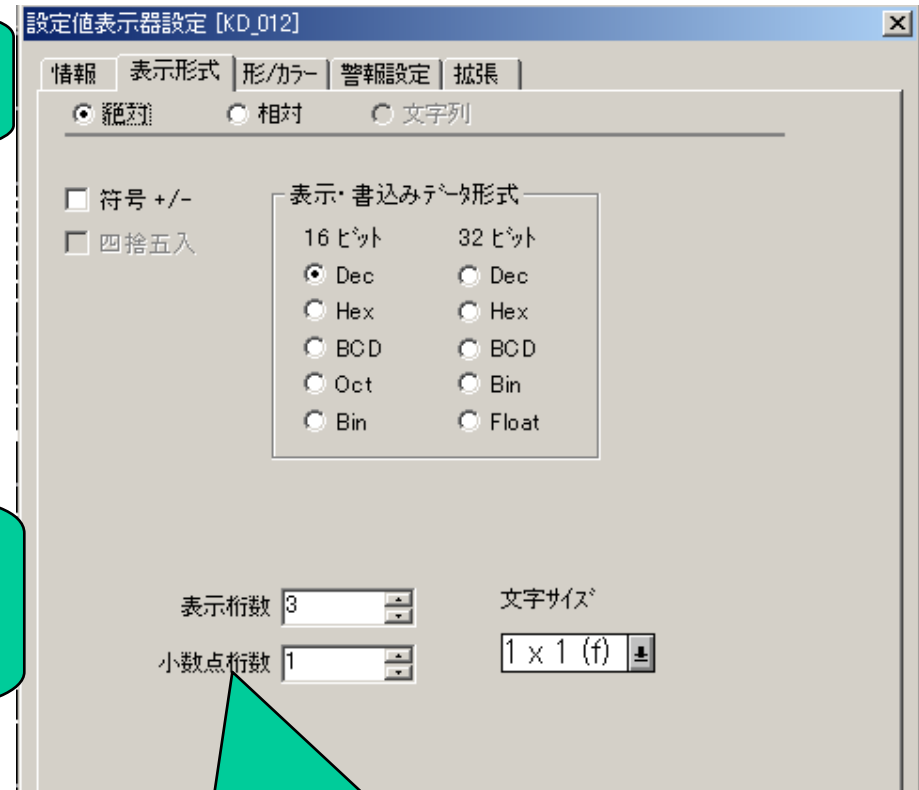
微分時間・・・設定値表示器設定



PID命令はどうやって使うの？

設定画面を作る！

無効範囲・・・設定値表示器設定



PID命令はどうやって使うの？

設定画面を作る！

オフセット・・・設定値表示器設定

PID命令はどうやって使うの？

グラフ表示画面を作る！

グラフ表示・・・折れ線グラフ設定
設定温度（SP）に対して、現在温度（PV）の経過を表示するグラフを作りましょう！

表示する数！

ノーマルを選択！

チャンネルごとに読み込むアドレスを設定する。

PID命令はどうやって使うの？

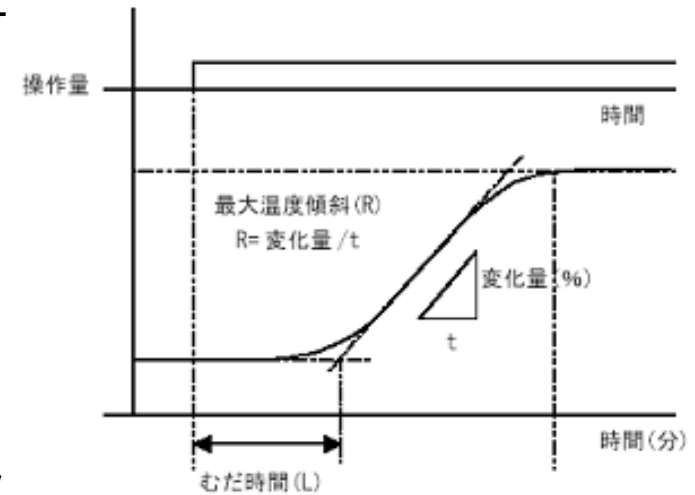
グラフ表示画面を作る！

設定温度 (SP)、現在温度 (PV)
設定画面で作成した部品を貼り付ける。

PID命令はどうやって使うの？

ステップ応答法

PID制御を行うにあたり、制御結果を最適なものにするために、ステップ応答法があります。
 算出方法は、目標値（PV）を設定し、出力値（CV）100%をステップ状に出力します。
 このときの右表の温度特性グラフより、最大温度傾斜（R）とむだ時間（L）を計測します。
 最大温度傾斜（R）とむだ時間（L）を計測した値を下記の方程式に代入して“比例係数”、“積分回数”、“微分時間”の定数を算出します。
 算出した値を「PID/チューニング」ダイアログボックスに代入してください。



$$\begin{aligned}
 \text{“比例係数”} &= 100 / (0.83 \cdot R \cdot L) \quad \text{【\%】} \\
 \text{“積分回数”} &= 1 / (2 \cdot L) \quad \text{【回/min】} \\
 \text{“微分時間”} &= 0.5 \cdot L \quad \text{【min】}
 \end{aligned}$$

注意事項

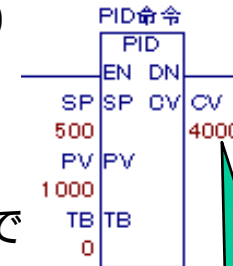
出力制限範囲と積分回数範囲の関係

下図のように設定したとします。

出力制限範囲とは、PID命令による演算にて、仮に-100ないし、4100という演算結果が出た場合、出力制限範囲と比較して、それが範囲外なら、-100という結果に対しては0、4100は4000と出力します。

そして、積分回数範囲とは、現在の出力値（CV）がここで設定した範囲内であれば、積分演算値を出力値に積算し、範囲外であれば積算しません。

．．．と、言葉で書いてもいまいち不明かもしれないので、次頁以降に図で説明致します。



この出力の
上下限範囲
を設定する。



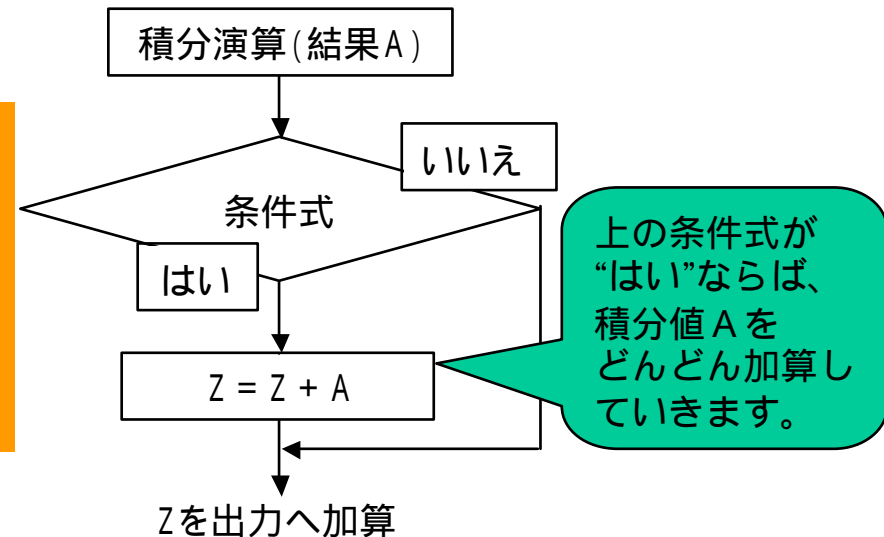
注意事項

出力制限範囲と積分回数範囲の関係

* 右図の条件式 *

以下の条件を1つでも満たすか？

1. 前回のPID命令の出力値が積分回数範囲内である。
2. 前回のPID命令の出力値が積分回数範囲上限以上及び積分値 $A < 0$ である。
3. 前回のPID命令の出力値が積分回数範囲下限以下及び積分値 $A > 0$ である。



$A = \text{偏差} \times \text{積分係数} \times \text{データ取得周期}$

* 積分係数 = 比例係数 / 1000 × (積分回数 / 6000000)

偏差 = SP - PV (正動作)、PV - SP (逆動作)

Zの範囲は、 - 2147483648 ~ 2147483647 です。

上図を説明します。

まず、積分回数範囲は下限値が - 1、上限値が 4001 とします。

(- 1 < 出力値 (CV) < 4001 となる)

PID命令の出力制限範囲を 0 出力値 (CV) 4000 だったとします。

この場合、出力値は最低でも 0、最高で 4000 となりますので、常に上記の条件式の “ 1 ” が成立する状態となります。(ずっと、“はい”になる)

よって、積分演算で求められた積分値Aの値をデータ取得周期毎に加算され続けます。

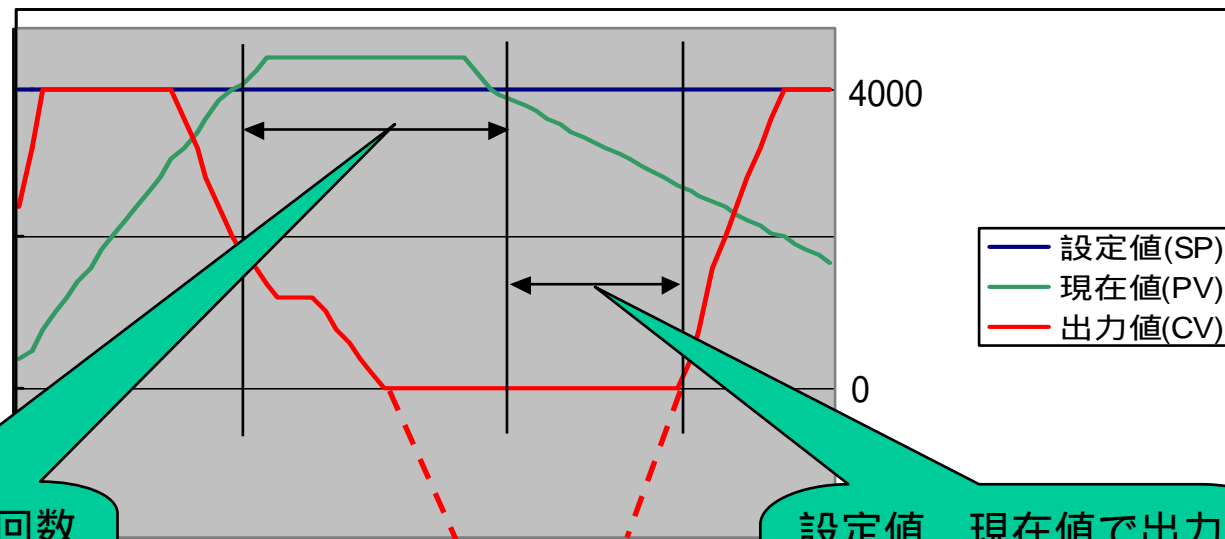
注意事項

出力制限範囲と積分回数範囲の関係

つまり、正動作にて、設定値 (SP) と現在値 (PV) が下図のグラフのようになったとき、この位置で、積分結果がデータ取得周期毎にマイナスの値を加算されていくので、内部での出力値は、大きくマイナス方向へ下がっていきます。

よって、設定値 現在値に変化しても、出力値は、マイナス方向から追従し始めるので、見た目では出力値が変わらず (追従せず)、制御遅れが発生します。

逆に設定値 現在値が長く続くと、プラスの値を加算 (プラス方向にて、上記と同じ現状を) していきます。



出力値が積分回数範囲以内なので、この間、積算される。

設定値 現在値で出力値は上がっていくが、マイナスからなので、追従が遅れる。

注意事項

出力制限範囲と積分回数範囲の関係

従って、このような制御遅れを無くすために、**出力制限範囲内で積分回数範囲を設定**して下さい。

先ほどまでの例で行くと、出力制限範囲が0 出力値 4000なので、積分回数範囲を下図のようにすれば、 $0 < \text{出力値} < 4000$ となります。

こうすると、出力が下限もしくは、上限に達したとき、積分結果を加算しないので、極度にプラスもしくはマイナス方向へいかなくなるので、追従性が早くなります。

